EXHIBIT B

FILED UNDER SEAL

YOUTUBE REMOTE (YTR) DOES NOT INVALIDATE THE '033 PATENT

Summary Judgment – Legal Standards

➢ Summary judgment is a "***lethal weapon*** and courts must be mindful of its aims and targets and ***beware of overkill in its use***."

Kangaroos U.S.A. v. Caldor, Inc., 778 F.2d 1571, 1574 (Fed. Cir. 1985) (emphasis added)

➢ Summary judgment is only appropriate "if 'there is ***no genuine issue*** as to any material fact and the moving party is entitled to judgment as a matter of law.'"

Vallavista Corp. v. Amazon.com, Inc., 657 F. Supp. 2d 1132, 1135 (N.D. Cal. 2008)
(quoting Fed. R. Civ. P. 56(c)) (emphasis added)

➢ Granting a motion for "summary judgment of noninfringement" "would require finding that ***no reasonable juror***, when drawing ***all reasonable inferences in favor of [patentee]***, could return a verdict of infringement."

Conceptus, Inc. v. Hologic, Inc., 771 F. Supp. 2d 1164, 1175 (N.D. Cal. 2010) (emphasis added)

Summary Judgment – Legal Standards

➢ When the summary judgment motion is "to invalidate an existing patent, ***the burden on the moving party is indeed heavy***."

Tillotson, Ltd. v. Walbro Corp., 831 F.2d 1033, 1036 (Fed. Cir. 1987) (emphasis added)

➢ And "a moving party seeking to invalidate a patent at summary judgment must submit such ***clear and convincing evidence*** of invalidity so that no reasonable jury could find otherwise."

Eli Lilly & Co. v. Barr Lab'ys, Inc., 251 F.3d 955, 962 (Fed. Cir. 2001) (emphasis added)

YTR Does Not Anticipate or Render Obvious the '033 Patent

➢ The record is replete with genuine disputes over material facts regarding what YTR discloses with respect to the asserted claims of the '033 Patent.

➢ A reasonable juror, especially when drawing all reasonable inferences in favor of Sonos, could return a verdict that the '033 Patent claims are valid over YTR.

➢ The Court should let the jury decide whether the '033 Patent is valid over YTR.

YTR Does Not Anticipate or Render Obvious the '033 Patent

➢ Because Google crammed 4 different summary judgment motions into a single filing, for the sake of brevity, Sonos has focused on YTR's lack of disclosure concerning limitation 1.7 of claim 1 of the '033 Patent.

➢ Limitation 1.7 is directed to a playback device taking over responsibility for playback of a remote playback queue.

➢ It should be understood that the same arguments apply to the corresponding limitation 12.4 in claim 12 of the '033 Patent.

Representative Claim 1 of '033 Patent – Relevant Portions

[1.0]   A computing device comprising:

[1.1]   at least one processor;

[1.2]   a non-transitory computer-readable medium; and

[1.3]   program instructions . . . comprising:

[1.4]   operating in a first mode in which the computing device is configured for playback of a remote playback queue provided by a cloud-based computing system associated with a cloud-based media service;

[1.5]   . . . displaying a representation of one or more playback devices . . . ;

[1.6]   . . . receiving user input indicating a selection of at least one given playback device . . . ;

[1.7]   **based on receiving the user input, [1.7(a)] transmitting an instruction for the at least one given playback device to take over responsibility for playback of the remote playback queue from the computing device, [1.7(b)] wherein the instruction configures the at least one given playback device to (i) communicate with the cloud-based computing system in order to obtain data identifying a next one or more media items that are in the remote playback queue, (ii) use the obtained data to retrieve at least one media item in the remote playback queue from the cloud-based media service; and (iii) play back the retrieved at least one media item;**

[1.8]   detecting an indication that playback responsibility for the remote playback queue has been successfully transferred from the computing device to the at least one given playback device; and

[1.9]   . . . transitioning from i) the first mode . . . to ii) a second mode . . . .

6

YTR Does Not Disclose Limitation 1.7 of the '033 Patent

Limitation 1.7: . . . [1.7(a)] transmitting an instruction for the at least one given playback device to take over responsibility for playback of the remote playback queue from the computing device, [1.7(b)] wherein the instruction configures the at least one given playback device to (i) communicate with the cloud-based computing system in order to obtain data identifying a next one or more media items that are in the remote playback queue . . .

➢ Limitation 1.7(a) requires configuring the "playback device" to "take over responsibility for playback of the remote playback queue."

➢ Limitation 1.7(b) requires configuring the "playback device" to "obtain data identifying a next one or more media items that are in the remote playback queue."

➢ As explained below, regardless of its mode of operation, the YTR System does not teach either (let alone both) of these subparts of limitation 1.7.

YOUTUBE REMOTE (YTR) OVERVIEW

YTR System – Party Mode Versus Non-Party Mode

➢ After using YTR's non-party mode to argue that YTR used a local playback queue that invalidates Claim 13 of the '615 Patent, Google now attempts to use YTR's party mode to argue that YTR used a remote playback queue that invalidates Claim 1 of the '033 Patent.

➢ But YTR party mode is merely an account extension of YTR non-party mode.

➢ YTR's ***non-party mode*** allowed multiple remote controls and multiple Leanback Screens to communicate with each other through a single MDx Server.

➢ YTR's ***party mode*** did the ***same*** thing – it allowed multiple remote controls and multiple Leanback Screens to communicate with each other through a single MDx Server.

➢ The ***only difference*** was that ***non-party mode*** remote controls and Leanback Screens had to be logged into the ***same YouTube Account***, while ***party mode*** purportedly permitted remote controls and Leanback Screens logged into ***different YouTube Accounts***.
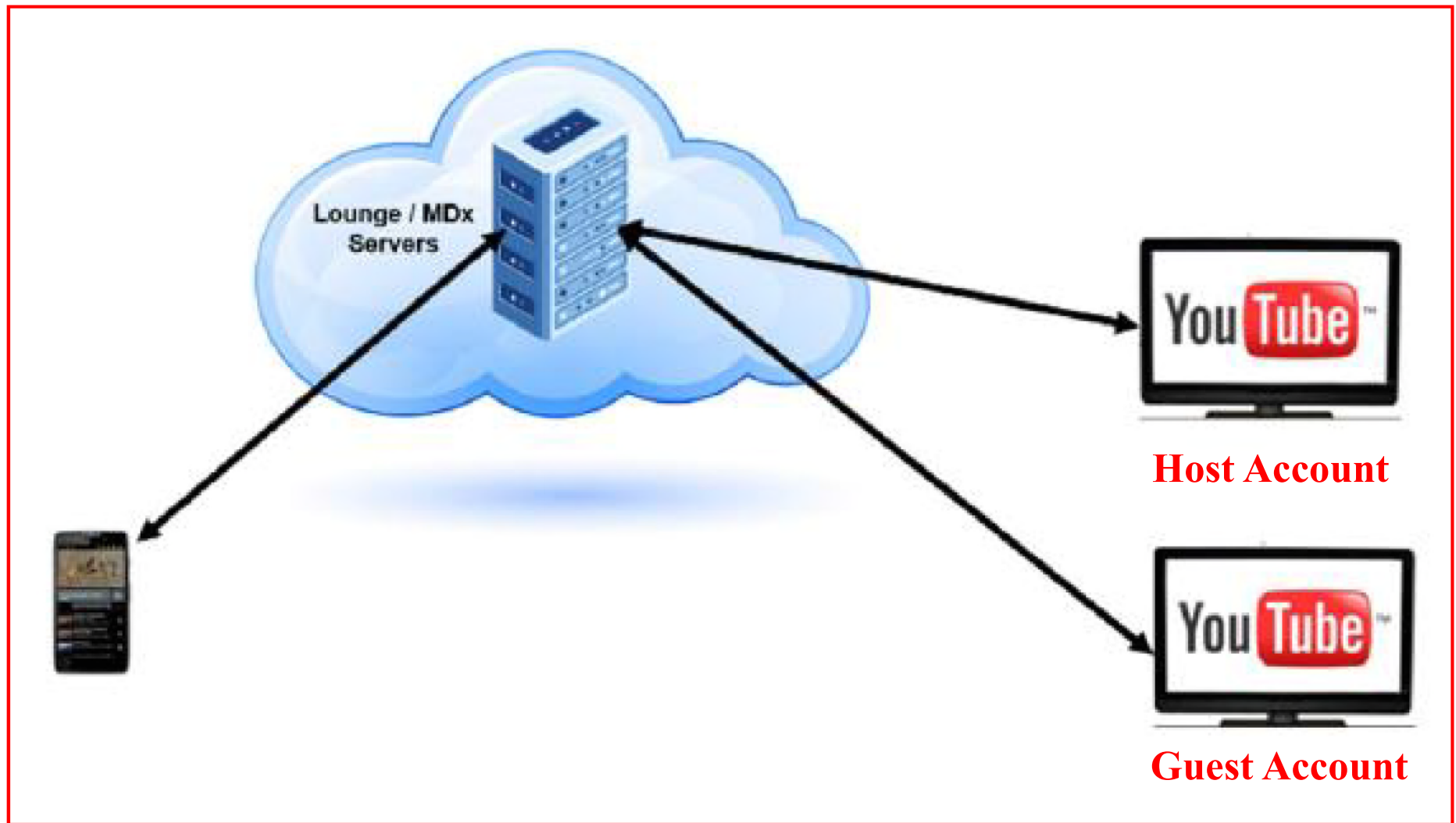
Sonos MSJ Opp. (Dkt. 509.02), pp. 4-5 (*citing* Ex. F (Dkt. 509.04), ¶199)

9

YTR System Architecture – Non-Party Mode

Sonos MSJ Opp. (Dkt. 509.02), Ex. F (Dkt. 509.04), ¶166 (annotated)

10
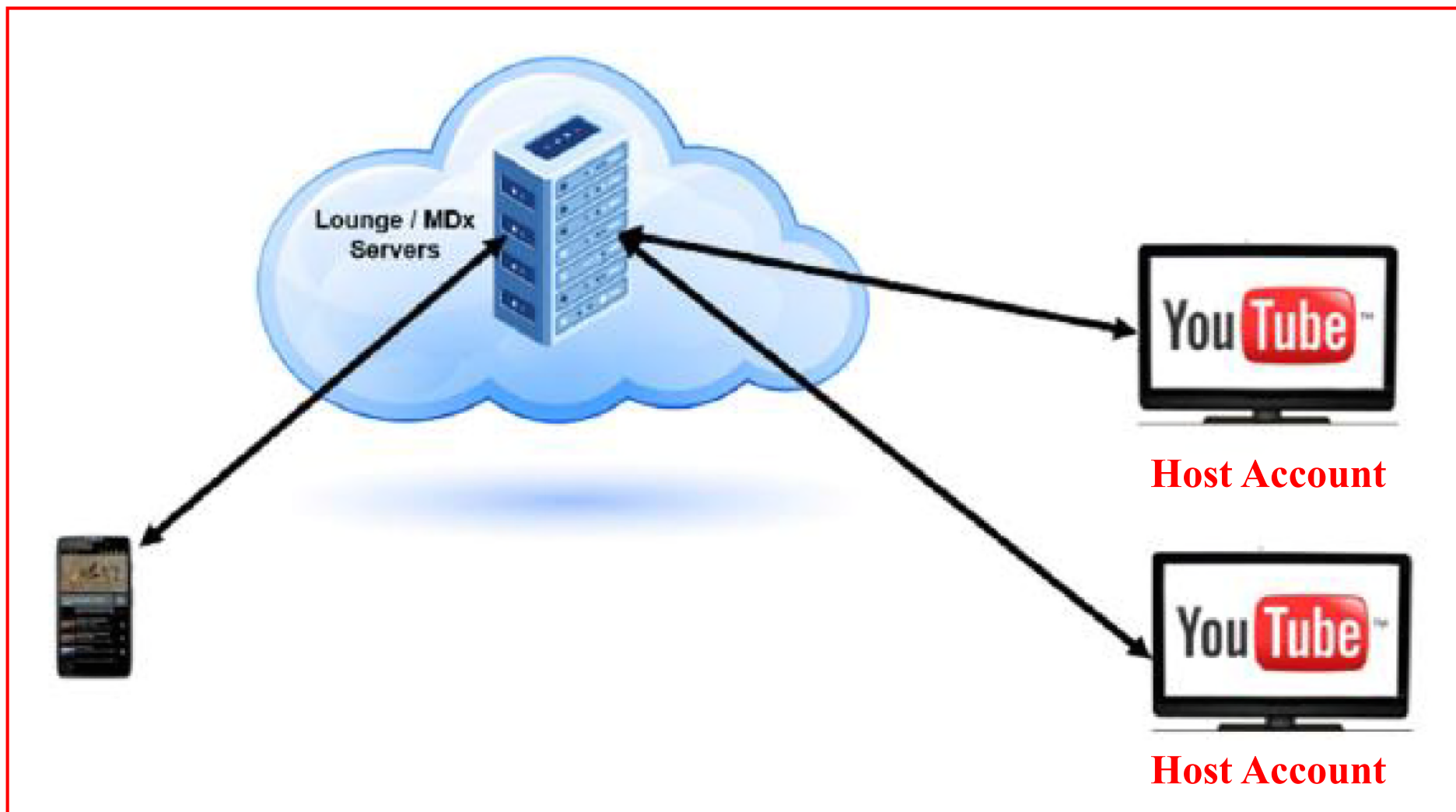
YTR System Architecture – Party Mode


Host Account

Guest Account

Sonos MSJ Opp. (Dkt. 509.02), Ex. F (Dkt. 509.04), ¶166 (annotated)

YTR System – Similarity Between Party and Non-Party Modes

➢ Party mode's ability to use TVs and smartphones logged in to different YouTube accounts means that the messages amongst the various devices in party mode inform the Lounge Server that it needs to relay the playlist and updates to not just the remote controls and Leanback Screens associated with the host's YouTube account, but also to the remote controls and Leanback Screens associated with any guests' YouTube accounts in the same session.

➢ However, in all other respects, the messages are the same for both non-party mode and party mode.
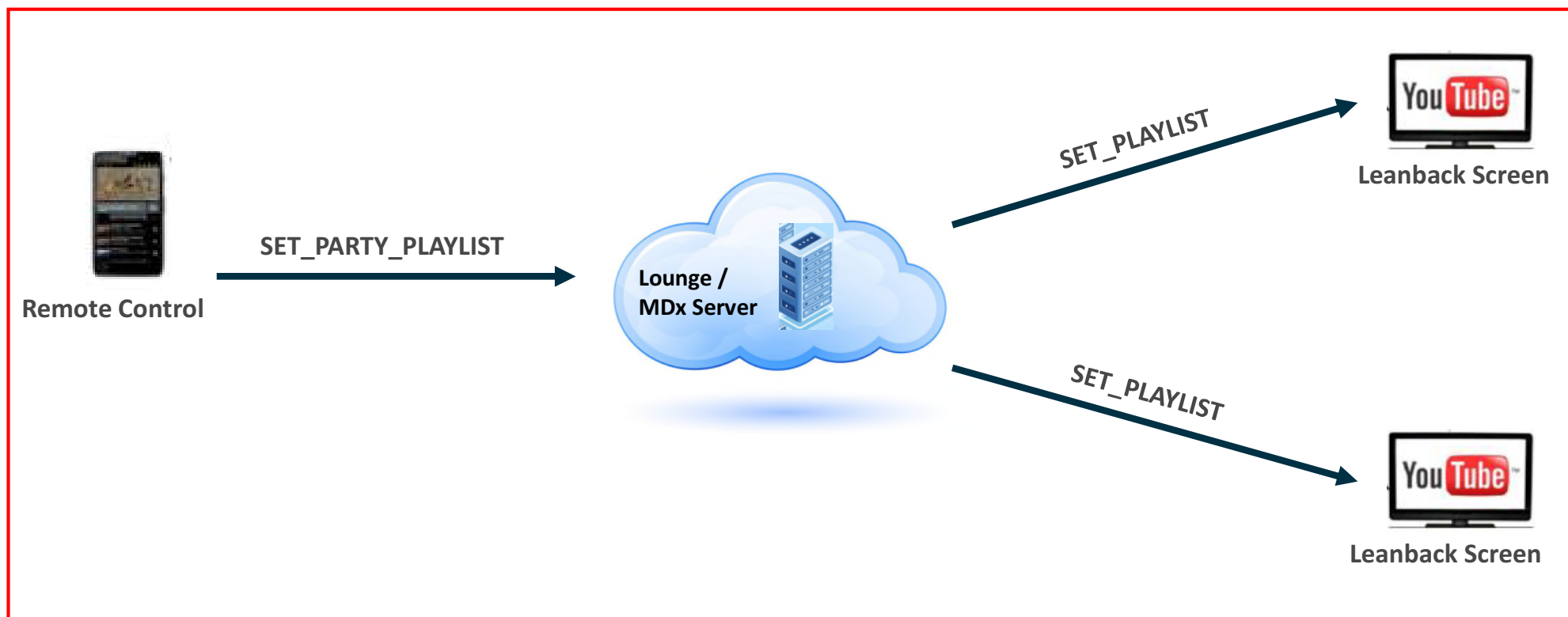
Sonos MSJ Opp. (Dkt. 509.02), pp. 5 (*citing* Ex. F (Dkt. 509.04), ¶200)

YTR System – Non-Party Mode Messages
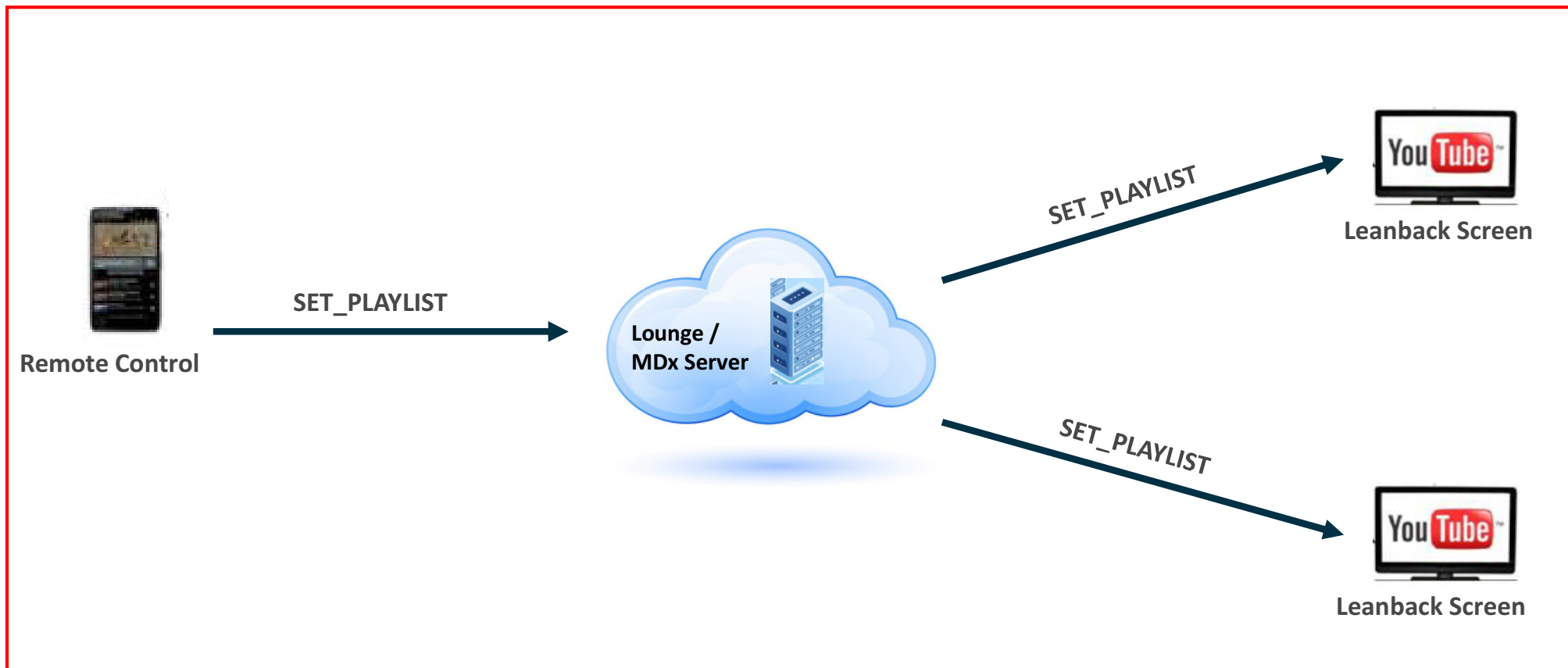
Demonstrative from Rebuttal Expert Report of Douglas C. Schmidt

13

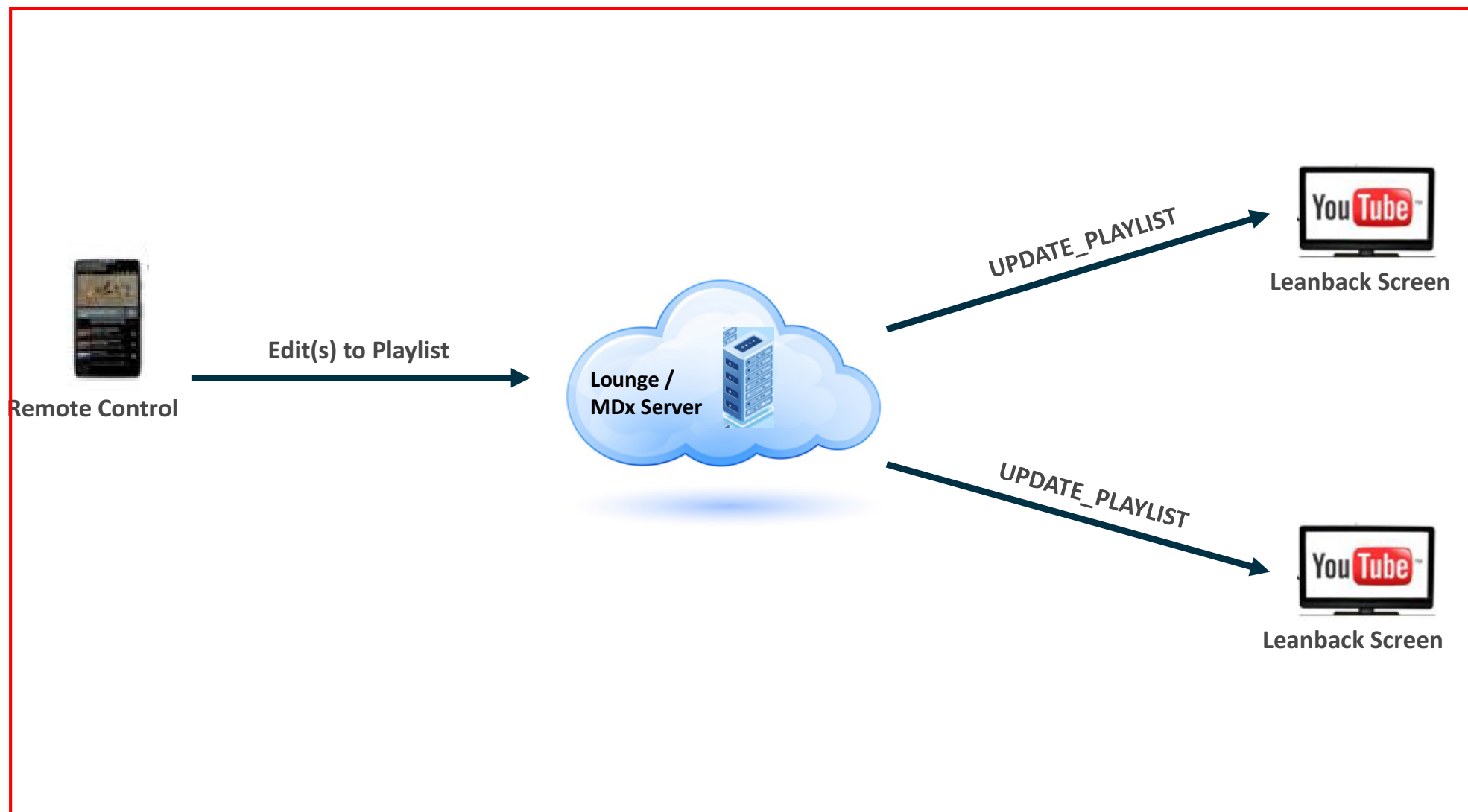# YTR System – Party Mode Messages

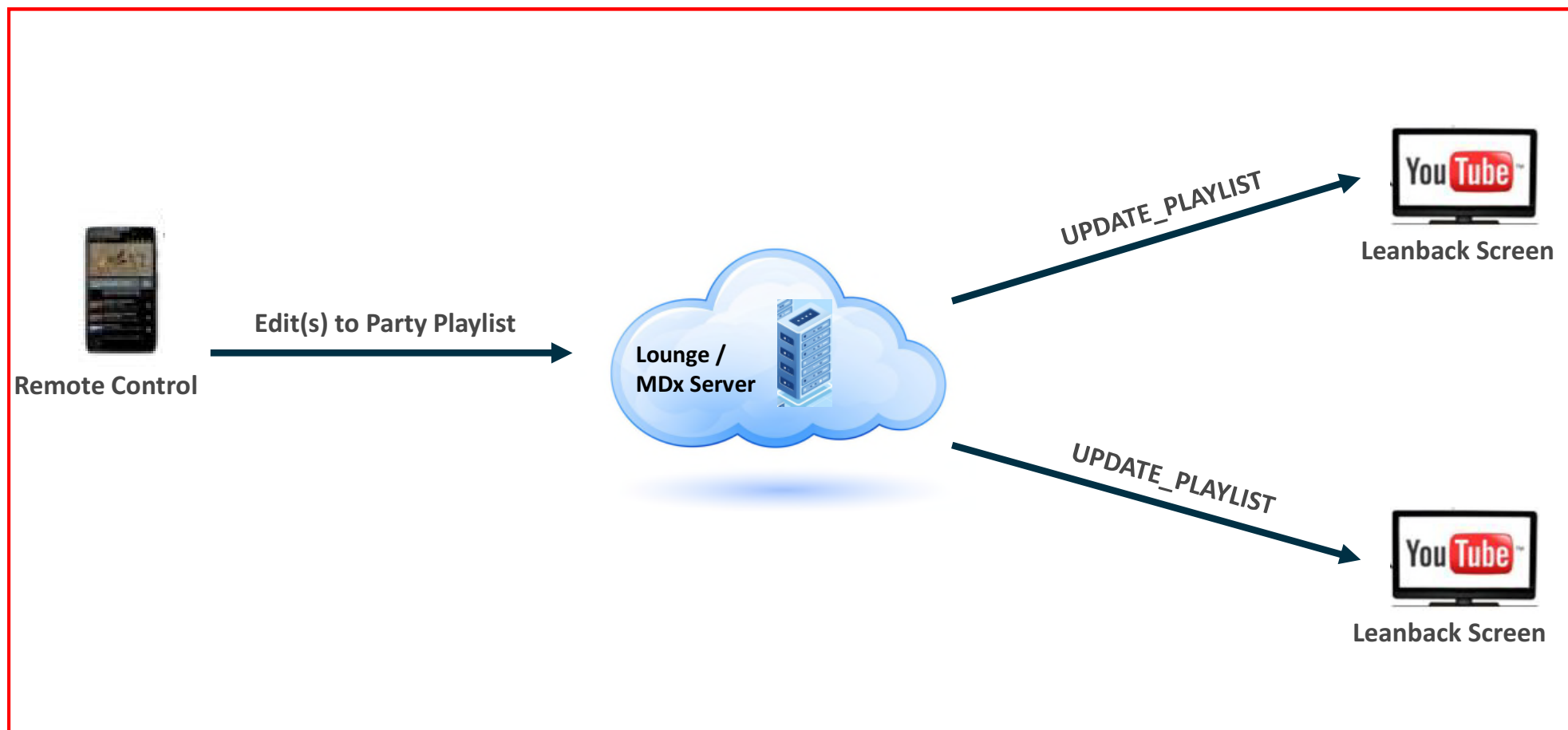Demonstrative from Rebuttal Expert Report of Douglas C. Schmidt

14

YTR System – Similarity Between Party and Non-Party Modes

➢ Google argues that YTR party mode uses a "remote playback queue" because "when the party queue is edited, the playback devices depend on the queue in the cloud to receive those edits." Reply, p. 4.

➢ But this argument doesn't hold water because the same thing happens when the queue is edited in non-party mode, which indisputably uses a "local playback queue."

➢ Just like in party mode, in non-party mode, any edits to the queue are sent by the MDx Server in the cloud to the playback devices.

Schmidt Reb. Report, ¶¶177, 213-214; Levai Dep. Tr. (January 6, 2023), 51:6-21, 52:5-14, 54:19-55:10, 55:11-18, 58:16-9, 59:15-60:12

15

YTR System – Non-Party Mode Messages

16

YTR System – Party Mode Messages

17

YTR Party Mode – Local Playback Queue Independent of MDx Server

➤ Google's technical expert, Dr. Bhattacharjee, further explained:

> That the accused YouTube system does not use a "local playback queue" is further evidenced by the fact that in the accused system if the MDx servers were to go offline, playback of the playlist would not be possible. In contrast, in a system that stored **the "playback queue" locally on the device, the playback device could continue to play the queue— whether a remote copy was available or not**. As an example, **consider the prior art YouTube Remote System** that I discuss in this Declaration (Section VII): **the playback device in the YouTube Remote prior art stores a list of videoIDs for the playlist and is thus capable of playing back the playlist even if the MDx server were not available**.

➤ This is exactly what happens in YTR party mode: the Screens maintain a local playback queue – regardless of "whether a remote copy was available or not" – and can continue playing videos even if the MDx Server goes offline.

Sonos MSJ Opp. (Dkt. 509.02), p. 6 (*quoting* Dkt. 210.03, ¶ 74)

May Contain Google Confidential Material

18

YTR Party Mode – Local Playback Queue Independent of MDx Server

Janos Levai
YTR Software
Engineer, Google

Q Well, all the devices such as the Leanback screens and the YouTube remotes, they also store and maintain a copy of that playlist; correct?

A Well, they need to so that they can show it in [GUI] to the user.

Q And so they can play it back; right?

A Well, in remote mode, yes, the lounge screen plays back those videos.

Q Plays back those videos stored locally on the screen; right?

A Well, it gets a list of videos and which one to play and starts playing that video. And then in order to know which video to play next when the current one ends, it refers to that list.
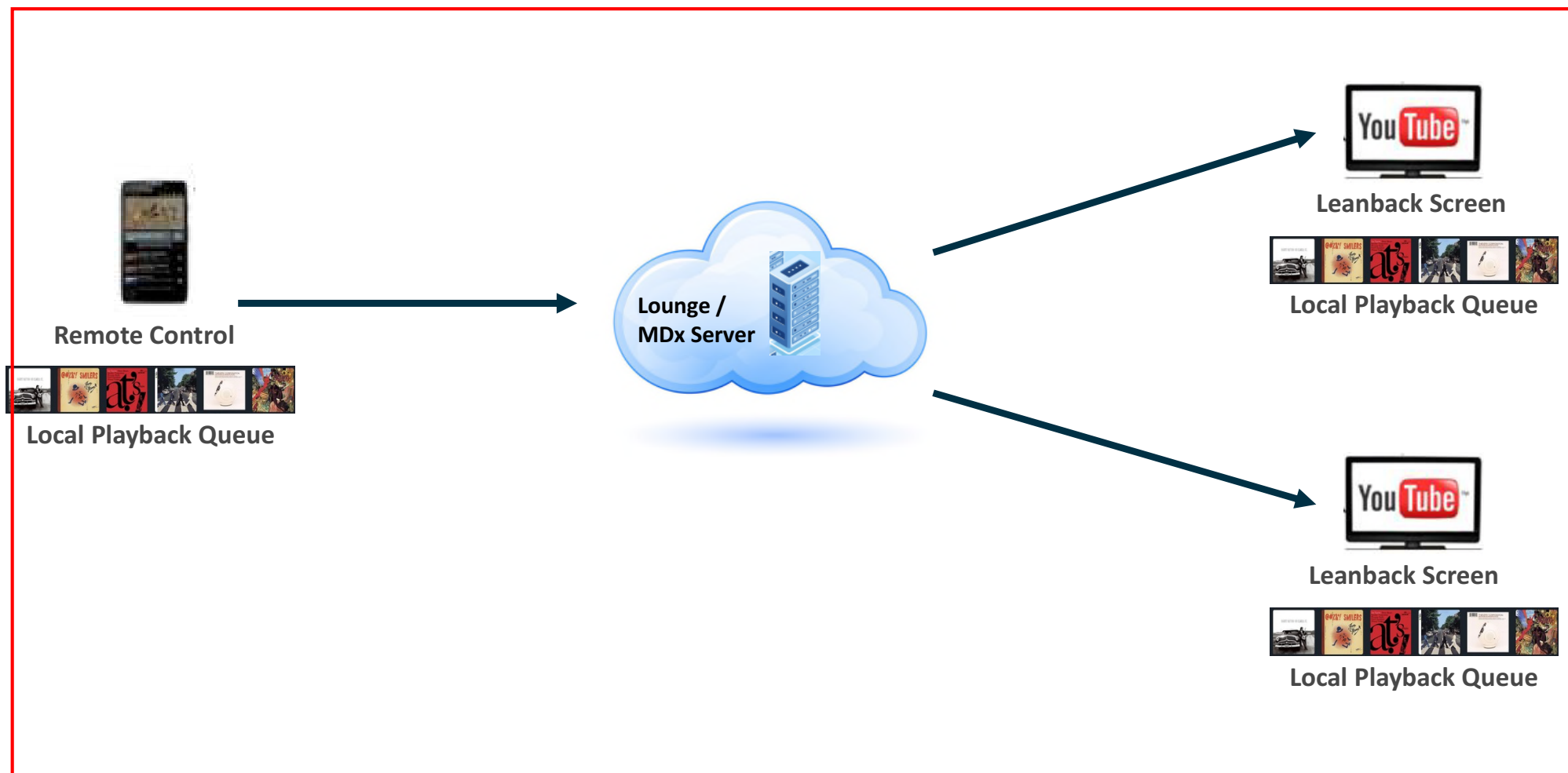
Q The locally stored list on the Leanback screens; correct?

A Yes. I don't believe it asked the MDx server for which video to play next when the current ended. I believe it referred to that list that it caught before.

Sonos MSJ Opp. (Dkt. 509.02), Ex. F (Dkt. 509.04), ¶210 (*quoting* Levai Dep. Tr. (January 6, 2023), 59:15-60:12)

19

YTR System – Local Playback Queue Independent of MDx Server

➢ Thus, according to Google's YTR software engineer, Mr. Levai, a Leanback Screen in party mode "starts playing that video" from its "*locally* stored list" of videos, "refers to that list" in order "to know which video to play next," and does *not* "ask[] the MDx server for which video to play next."

➢ Put another way, in either YTR party mode or non-party mode, the local playback queue get a copy of the *entire* playlist and does *not* depend on any playlist maintained by the MDx Server.

Sonos MSJ Opp. (Dkt. 509.02), p. 7

YTR System – Local Playback Queues In Non-Party Mode

Remote Control

Local Playback Queue

**Lounge /
MDx Server**

Leanback Screen

Local Playback Queue

Leanback Screen

Local Playback Queue

Demonstrative from Rebuttal Expert Report of Douglas C. Schmidt

21

YTR System – Local Playback Queues In Party Mode

Remote Control

Local Playback Queue

Lounge / MDx Server

Party Playlist

Leanback Screen

Local Playback Queue

Leanback Screen

Local Playback Queue

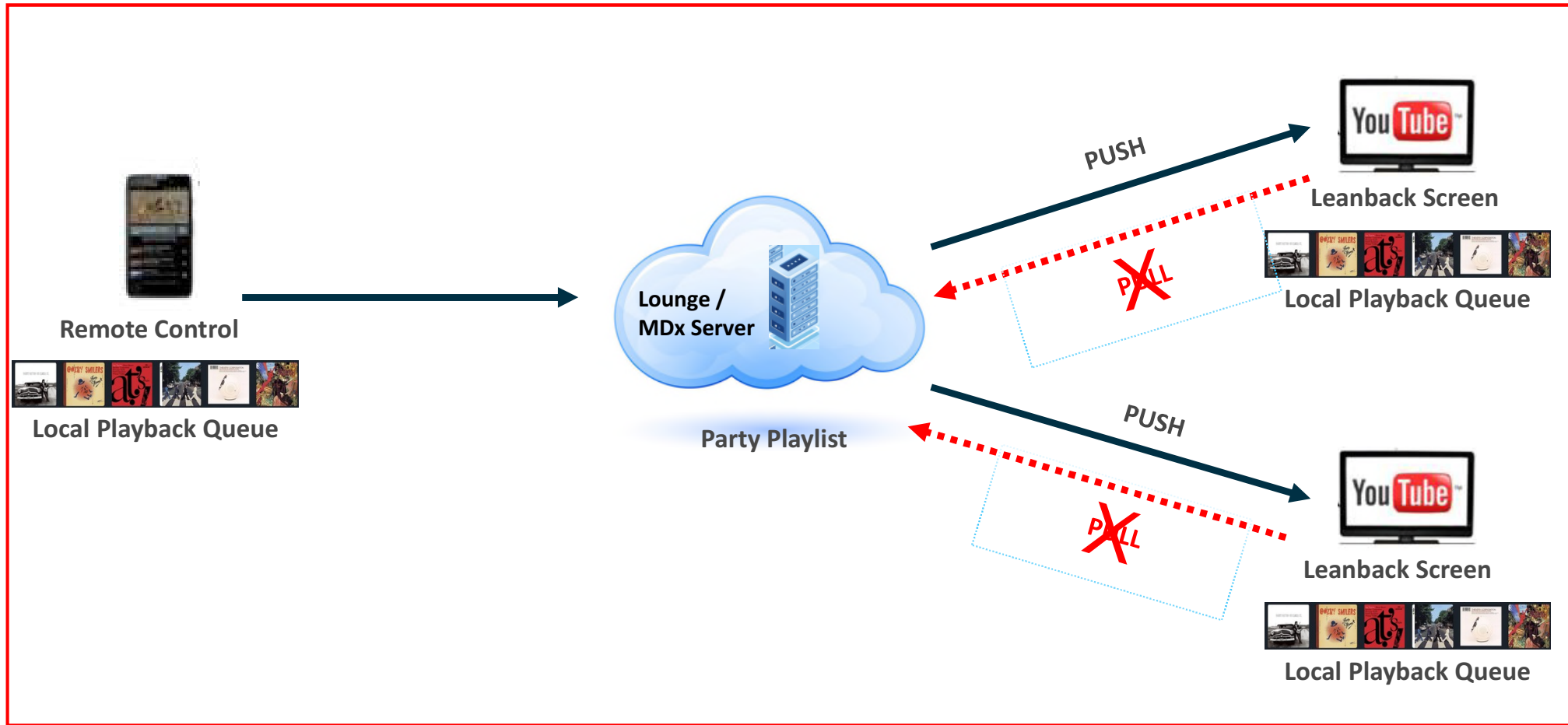Demonstrative from Rebuttal Expert Report of Douglas C. Schmidt

22
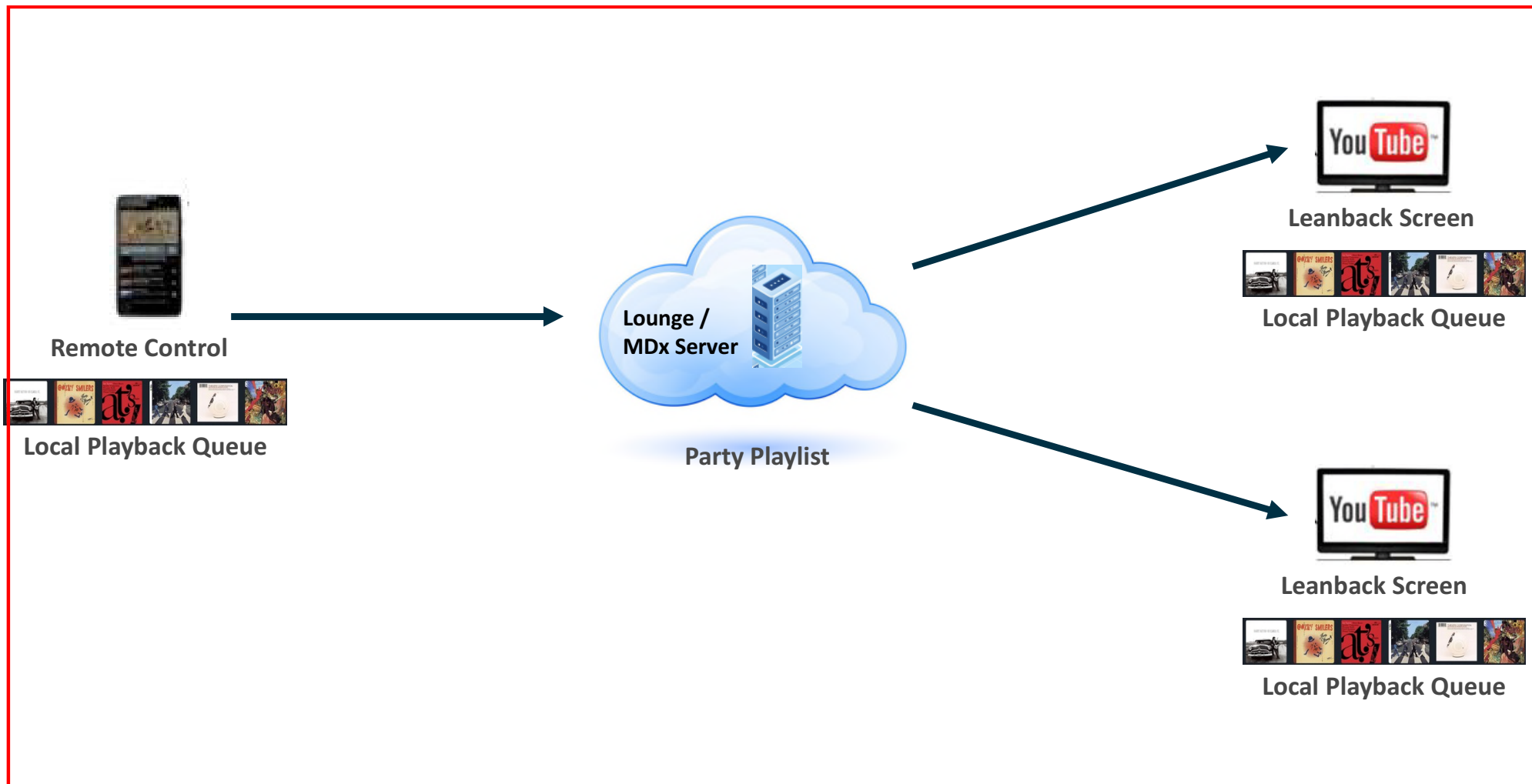
YTR System – Local Playback Queue Does Not Pull from MDx Server

➢ In arguing that the '033 Patent does not disclose a remote playback queue, Google's technical expert, Dr. Bhattacharjee, opined that a "***pull***" of media items is associated with a ***remote playback queue***, while a "***push***" of media items is associated with a ***local playback queue***:

> While the proxy server is in the cloud, there is no disclosure in the patent that the local playback system is configured to communicate with the cloud-based computing system after receiving the claimed instruction. Instead, the patent at most suggests that the ***one or more URLs are "pushed" to the local playback [system]*** from the third-party application via the proxy server when transfer is initiated. Id. There is ***no disclosure of the local playback system*** receiving the claimed instruction and ***performing a "pull" in order to obtain data identifying a next one or more media items that are in the remote playback queue***.

➢ Same for YTR: videoIDs are "pushed" from the MDx Server to the Screens – the Screens do not "pull" videoIDs from the MDx Server.

➢ Under Google's logic, that means YTR does not use a remote playback queue.

Opening Expert Report of Samrat Bhattacharjee Regarding Invalidity of [the '033 Patent], ¶714

23

YTR System – Local Playback Queues In Party Mode

Demonstrative from Rebuttal Expert Report of Douglas C. Schmidt (annotated)

24

THE MEANING OF "PLAYBACK QUEUE"

Judicial Estoppel

➢ Google tries to avoid estoppel by focusing on the prior invalidity issues before the Court regarding the '615 Patent while ignoring the noninfringement arguments that Google made. Reply, pp. 1-2.

➢ But Google is estopped from arguing a different meaning for "playback queue" because of what Google asserted for noninfringement, *not* invalidity.

➢ For purposes of summary judgment in the Patent Showdown, the parties did not dispute whether YTR had a local playback queue.

Judicial Estoppel

➢ Google distinguishes (Reply, p. 1) the Patent Showdown summary judgment as being related to YTR1, while this summary judgment relates to YTR2.

➢ But Google relied on both YTR1 and YTR2 in the Patent Showdown – it just conveniently hid the fact that YTR2 included a party mode.

Changelog for YouTube Remote

Jan 26, 2012	UPDATE	Version 3.1.0
Jan 13, 2012	UPDATE	Version 3.0.1
Aug 10, 2011	UPDATE	Version 2.0.7
Jul 29, 2011	UPDATE	Version 2.0.3
Jun 26, 2011	MORE DOWNLOADS	>250,000 downloads
Jan 12, 2011	UPDATE	Version 1.4
Jan 6, 2011	UPDATE	Version 1.3
Dec 24, 2010	MORE DOWNLOADS	50,000-250,000 downloads
Dec 17, 2010	UPDATE	Version 1.2
Dec 4, 2010	UPDATE	Version 1.1

Opening Expert Report of Samrat Bhattacharjee for Patent Showdown, ¶¶169-171

27

Judicial Estoppel

➤ As Google noted (Reply, p. 3), the Court construed "playback queue" to mean "a list of multimedia content selected for playback." Dkt. 316, p. 5.

➤ In its summary judgment decision, however, the Court further construed "playback queue" based on Google's noninfringement arguments.

➤ That the Court further construed the term "playback queue" is evident from the fact that the accused YT System indisputably had "a list of multimedia content selected for playback" stored locally on the accused playback devices – "the last played, the current one, and the next one."

THE COURT: Now, as I read the paperwork, what you do have is *the last played, the current one, and the next one*?

MR. VERHOEVEN [FOR GOOGLE]: Right.

Dkt. 308, 7/13/22 Hr'g Tr., 42-43

28

Judicial Estoppel

➢ Thus, for there to be no infringement, "playback queue" must have meant more than just "a list of multimedia content selected for playback."

➢ In fact, Google argued that it meant a lot more.

➢ The Court adopted Google's arguments.

➢ Google cannot change its tune now.

Google Told the Court There Can Be Only One Queue

➢ Google also told the Court that the existence of a "local playback queue" in a system is ***mutually exclusive*** of a "remote playback queue" (or "cloud queue") and ***vice versa***:

THE COURT: All right. So, Mr. Verhoeven, ***the argument against you here is that -- there are two queues: There is one in the Cloud, and there is another one on the speaker***. The local playback queue is on the speaker, and all it needs to know is what is the next one. And so that's good enough for its purposes and just calls the next one. . . . What do you say to that?

MR. VERHOEVEN [FOR GOOGLE]: I say that ***that's unpersuasive in the extreme*** . . . We're talking about <u>***a***</u> queue. ***Somebody has to own that queue***. Somebody has to ***maintain that queue***. Somebody has to be ***in charge of that queue***. Who is it? It's the Cloud. That's why it's called ***the*** Cloud ***queue***. ***The queue*** is maintained in the Cloud. If you want to know -- if you're the speaker and you want to know the next item in ***the queue***, you ask the Cloud because the Cloud maintains ***the queue***.

Sonos MSJ Opp. (Dkt. 509.02), pp. 2-3 (*quoting* Dkt. 308, 7/13/22 Hr'g Tr., at 59:16-60:9)

30

Google Told the Court There Can Be Only One Queue

THE COURT: Now, as I read the paperwork, what you do have *is the last played, the current one, and the next one*?

MR. VERHOEVEN [FOR GOOGLE]: Right.

THE COURT: Why isn't that enough to satisfy the queue?

MR. VERHOEVEN [FOR GOOGLE]: Because *that's not a queue*.

THE COURT: Why isn't it?

MR. VERHOEVEN [FOR GOOGLE]: *That's the processing of the queue* that's resident on the Cloud.

Dkt. 308, 7/13/22 Hr'g Tr., 42-43

The Court Relied On Google's Prior Representations

➢ Based on Google's representations, the Court indicated in its summary judgment decision for the '615 Patent (Dkt. 316) that:

• A "playback queue" must have a complete list of all the multimedia items that are to be played back – a "subset" or "short list" of such items is not enough to be a "playback queue." *Id*., 9 ("The groups of three items stored by the respective apps are not lists of multimedia content selected for playback."), 10 ("The passage thus distinguishes a local playback queue from the 'short list of tracks.'").

• A queue was not a "playback queue" if it "merely provide[d] the means to *process* the list[] for playback." *Id*., 10 (emphasis in original).

• "In short," the "playback queue" is the "queue [that] runs the show." *Id*.

YTR DOES NOT DISCLOSE LIMITATION 1.7

YTR System Fails to Disclose Limitation 1.7 of the '033 Patent

➢ Limitation 1.7 reads as follows:

[1.7] based on receiving the user input, **[1.7(a)]** transmitting an instruction for the at least one given playback device to take over responsibility for playback of the remote playback queue from the computing device, **[1.7(b)]** wherein the instruction configures the at least one given playback device to **(i)** communicate with the cloud-based computing system in order to obtain data identifying a next one or more media items that are in the remote playback queue, **(ii)** use the obtained data to retrieve at least one media item in the remote playback queue from the cloud-based media service; and **(iii)** play back the retrieved at least one media item

➢ Neither YTR party mode nor YTR non-party mode discloses this limitation and its subparts.

➢ For the same reasons, YTR does not disclose limitations 1.8-1.9, which build on limitation 1.7.

Sonos MSJ Opp. (Dkt. 509.02), pp. 7-10

YTR System Does Not Disclose Limitation 1.7(a)

➢ Limitation 1.7(a) reads: *transmitting an instruction for the at least one given playback device to take over responsibility for playback of the remote playback queue from the computing device.*

➢ The YTR Leanback Screens – the alleged "playback devices" – use a local playback queue regardless of which mode they are in.

➢ As such, they do not "take over responsibility for playback of the **remote playback queue** from the computing device."

Sonos MSJ Opp. (Dkt. 509.02), p. 8

35

YTR System Does Not Disclose Limitation 1.7(a)

➢ Google's argument (Reply, p. 5) that YTR's use of "service-recommended videos" constitutes a "remote playback queue" is misplaced.

➢ Google only relies on "service-recommended videos" for limitation 1.4, *not* limitation 1.7.

➢ The '033 Patent claims require **both** the remote control (limitation 1.4) and the playback devices (limitation 1.7) to use a remote playback queue:

> **[1.4] . . . the *computing device* is configured for playback of a *remote playback queue* . . . ;**

> **[1.7] . . . the at least one given *playback device* to take over responsibility for playback of the *remote playback queue* from the computing device . . .**

36

YTR System Does Not Disclose Limitation 1.7(b)

➢ Limitation 1.7(b)(i) reads: *wherein the instruction configures the at least one given playback device to (i) communicate with the cloud-based computing system in order to obtain data identifying a next one or more media items that are in the remote playback queue.*

➢ Google argues that YTR discloses subpart (i) because the Leanback Screens "sent a message to the YouTube 'Player Service' in the cloud to obtain one or more URLs (called 'Bandaid URLs') that were used to retrieve the audio and video content of the next video in the party queue from Google's Content Delivery Network (called 'Bandaid')." Google's Mot. at 11.

➢ This argument is flawed for at least three reasons.

Sonos MSJ Opp. (Dkt. 509.02), p. 8

YTR System Does Not Disclose Limitation 1.7(b)

➢ ***First***, the claim language in Limitation 1.7(b)(i) that "***data identifying*** a next one or more ***media items***" does not have to be a URL, but rather, could be another type of identifier, such as a videoID.

➢ The claim plainly does not refer to a URL.

➢ The '033 Patent's specification makes clear that, in addition to a URL, the claimed "data" may be "an identifier" or "some other identification," such as a "song identifier." *See, e.g.*, '033 Patent at 12:50-58 ("a URL (***or some other identification*** or address) for a song and/or playlist"); 15:55-63 ("***an identifier*** for a single track, a playlist, a streaming radio station, a programmed radio station, and so on"); 13:33-37 ("an application can pass a ***song identifier*** to a local playback system [to] find[] an available audio stream").

Sonos MSJ Opp. (Dkt. 509.02), p. 8
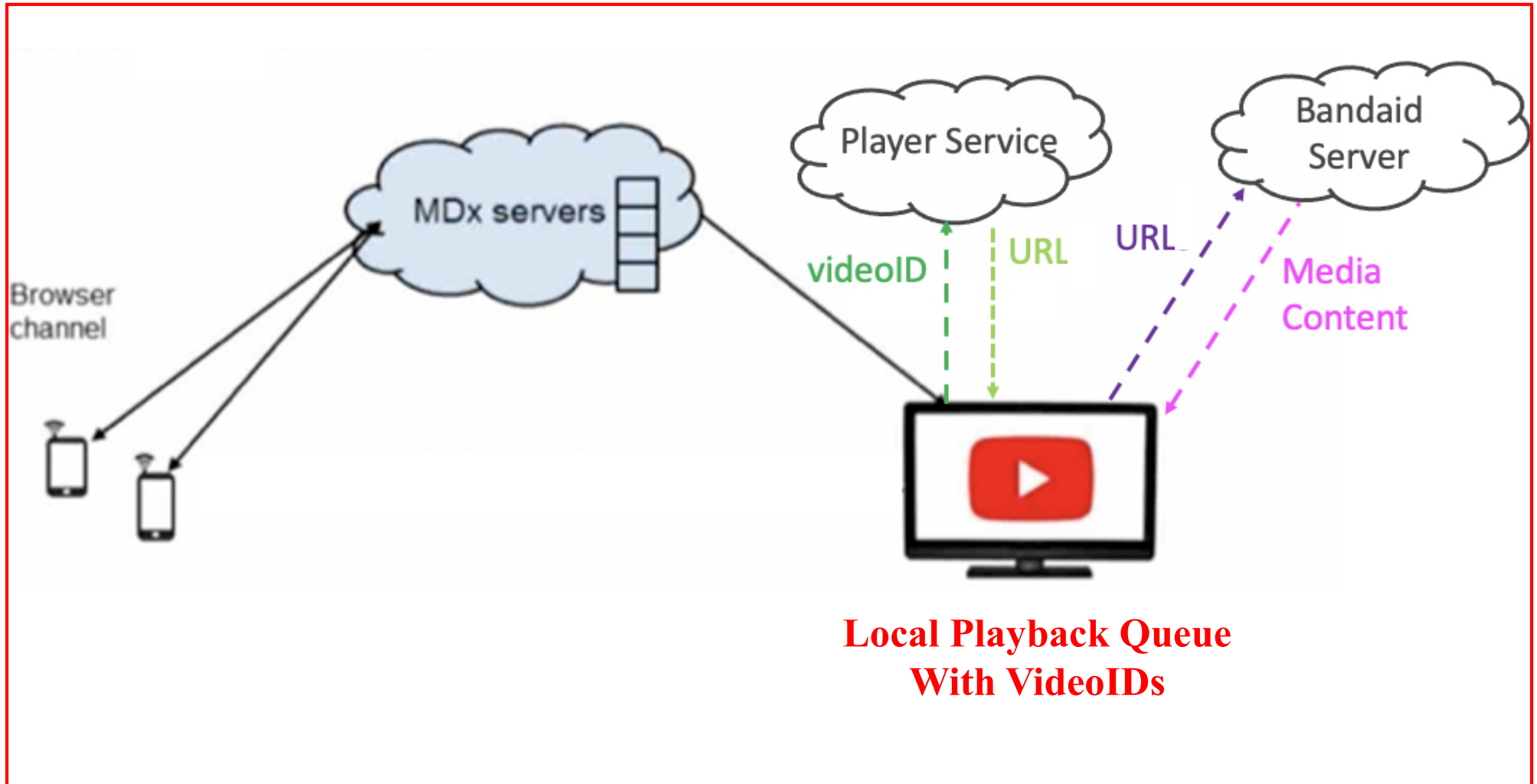
YTR System Does Not Disclose Limitation 1.7(b)

➢ Google now agrees that "data identifying . . . media items" may be either a videoID *or* a URL. Reply, p. 6 n. 5.

➢ However, Google argues that in YTR such data is only a Bandaid URL, and not the videoID. Reply, p. 6.

➢ Sonos disputes this conclusion, as the videoID is what resides in the playback queue and is what the playback device uses to obtain the media content located at the Bandaid URL.

➢ In fact, the Bandaid URL cannot be obtained without the videoID.

Sonos MSJ Opp. (Dkt. 509.02), p. 8

39

YTR System Does Not Disclose Limitation 1.7(b)

➢ ***Second***, limitation 1.7(b)(i) requires that the "data identif[ies] a ***next*** one or more media items."

➢ In YTR, the Bandaid URLs Google points to only identify the ***current*** media item for playback, ***not*** the ***next*** media item.

➢ The Leanback Screen sends a videoID for a video from its local playback queue to the Player Service, which provides a corresponding Bandaid URL for that videoID.

➢ The Leanback Screen then uses that Bandaid URL to obtain the video content from the Bandaid server and play it back.

➢ Therefore, the Leanback Screens use the videoIDs and corresponding Bandaid URLs to obtain the video content that it is ***currently*** set to play back, ***not*** the video that is set to play back ***next***.

Sonos MSJ Opp. (Dkt. 509.02), p. 9 (*citing* Ex. F (Dkt. 509.04), ¶¶176, 212; Ex. E (Dkt. 509.03), ¶¶162-166)

40

YTR System Does Not Disclose Limitation 1.7(b)

**Local Playback Queue
With VideoIDs**

Ex. E (Dkt. 509.03), ¶166 (annotated)

41

YTR System Does Not Disclose Limitation 1.7(b)

➢ Google's argument (Reply, p. 6) that Sonos's expert, Dr. Schmidt, provided an opinion that "Bandaid URLs" constitute "data identifying a next one or more media items" is highly misleading.

➢ Google cites to Paragraph 502 of Dr. Schmidt's opening report, which is directed to the infringement of Google's NIA #3.

Alleged Alternative #3

➢ NIA #3 is a new design that has not been commercially released yet and has absolutely nothing to do with how the prior art YTR operated.

Sonos MSJ Opp. (Dkt. 509.02), p. 9 (*citing* Ex. F (Dkt. 509.04), ¶¶176, 212; Ex. E (Dkt. 509.03), ¶¶162-166)

YTR System Does Not Disclose Limitation 1.7(b)

➢ Google's assertion (Reply, p. 6) that there is "evidence showing that a playback device **also** obtains Bandaid URLs for each of the next videos in the party queue (Video B, Video C, etc.)" is highly misleading as well.

➢ As Google's expert, Dr. Bhattacharjee, explained "for **each** item of media to be played back, the [] player issues a [] request to obtain **a** Bandaid URL." Dkt. 482-6 ¶¶ 328-329.

➢ Thus, Bandaid URLs are obtained **one at a time** for the **current** video – YTR does not access multiple Bandaid URLs for the current **and** a next video at the same time.

43

YTR System Does Not Disclose Limitation 1.7(b)

➢ Specifically, when its Video A's time to be the current video for playback, the videoID for Video A is sent to the Player Service to obtain a corresponding Bandaid URL for Video A.

Ex. E (Dkt. 509.03), ¶166 (annotated)

44

YTR System Does Not Disclose Limitation 1.7(b)

➢ And when Video A is done playing, and its Video B's time to be the current video for playback, the videoID for Video B is sent to the Player Service to obtain a corresponding Bandaid URL for Video B, and so on.
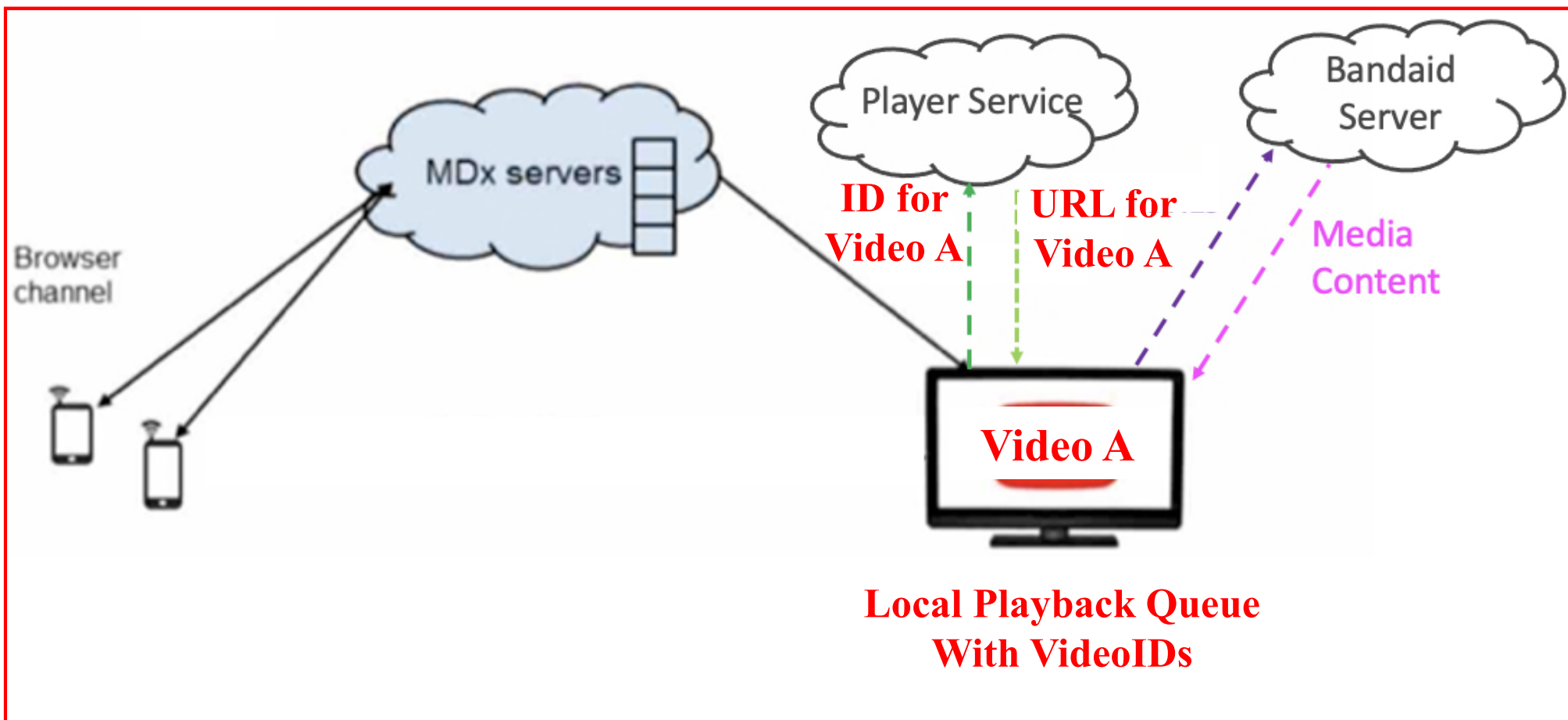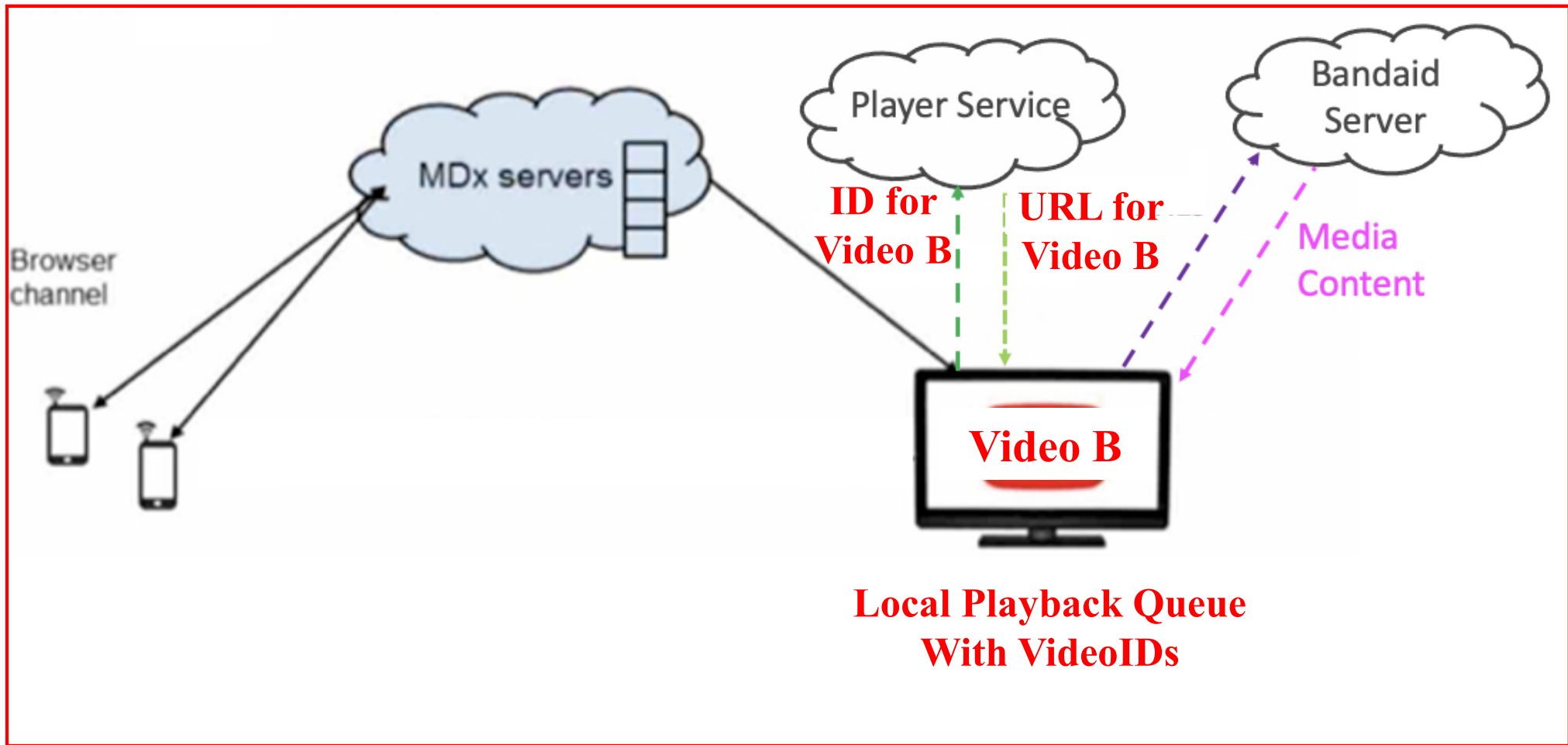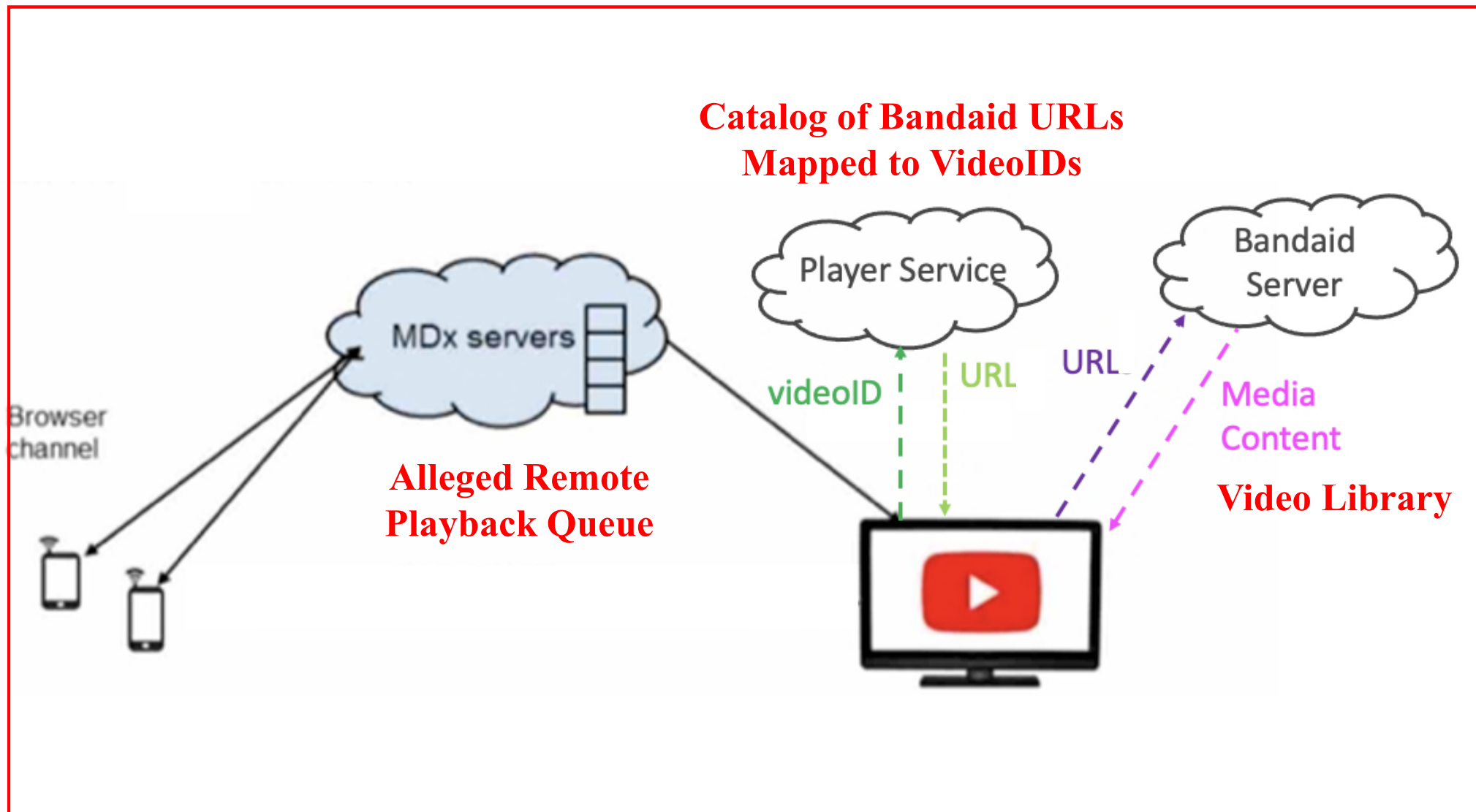

Ex. E (Dkt. 509.03), ¶166 (annotated)
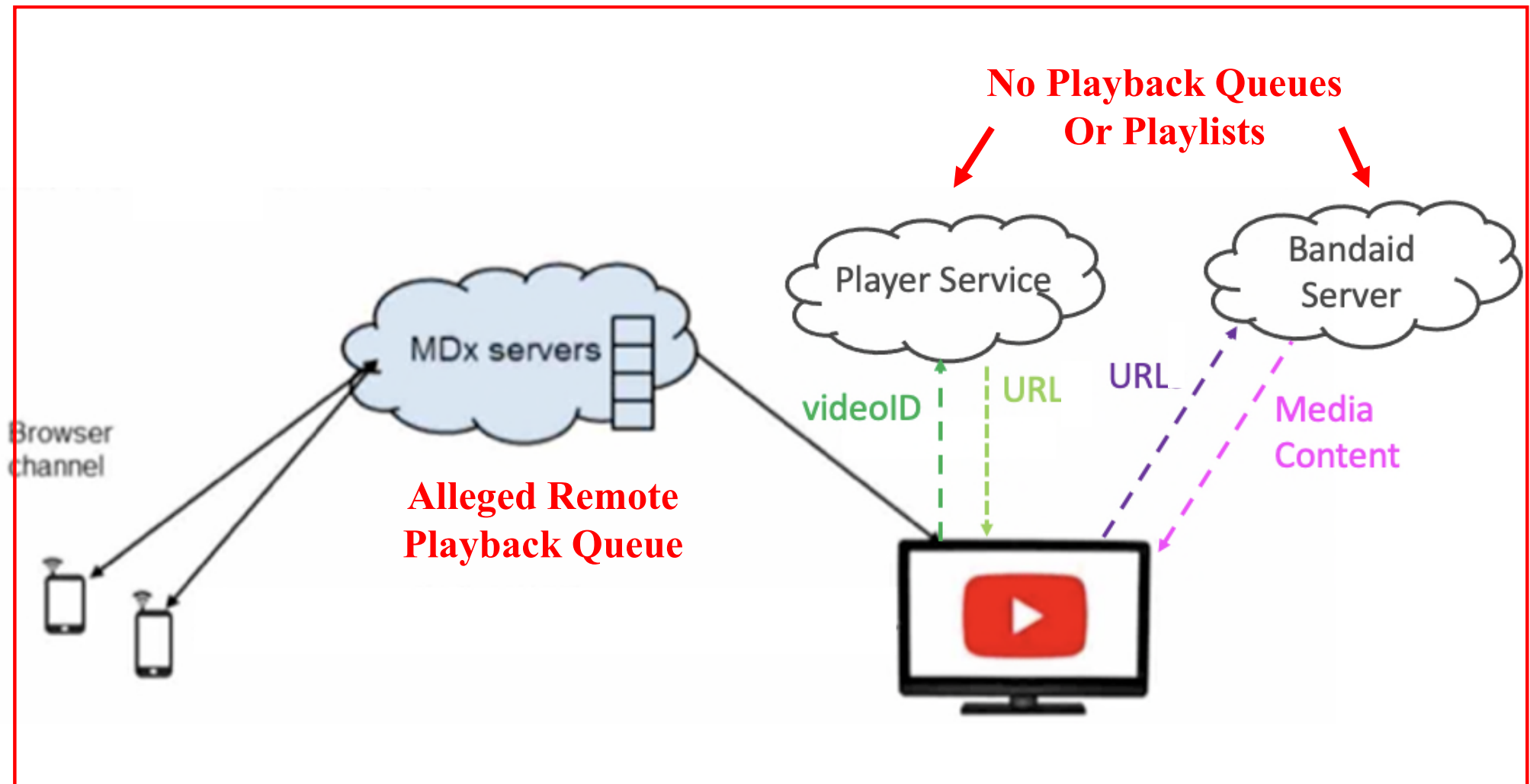
YTR System Does Not Disclose Limitation 1.7(b)

➢ ***Third***, limitation 1.7(b)(i) also requires that the playback device "obtain data identifying a next one or more media items that are ***in the remote playback queue*.**"

➢ In either YTR party mode or non-party mode, each Leanback Screen gets such data from its local playback queue, not a remote playback queue.

➢ As explained by Mr. Levai, the Leanback Screens in YTR party mode do not ask the MDx Server for "which video to play next when the current [video] ended."

➢ Neither the "Player Service" nor the "Bandaid" server maintains any YTR playback queues or playlists – they simply provide mapping and content services.

Sonos MSJ Opp. (Dkt. 509.02), pp. 9-10 (*citing* Ex. E (Dkt. 509.03), ¶¶163-66, Ex. F (Dkt. 509.04), ¶¶170, 204, 201)

46

YTR System Does Not Disclose Limitation 1.7(b)

**Catalog of Bandaid URLs
Mapped to VideoIDs**

**Alleged Remote
Playback Queue**

Video Library

Ex. E (Dkt. 509.03), ¶166 (annotated)

47

YTR System Does Not Disclose Limitation 1.7(b)

No Playback Queues Or Playlists

Player Service

Bandaid Server

MDx servers

videoID   URL   URL   Media Content

Browser channel

Alleged Remote Playback Queue

Ex. E (Dkt. 509.03), ¶166 (annotated)

48

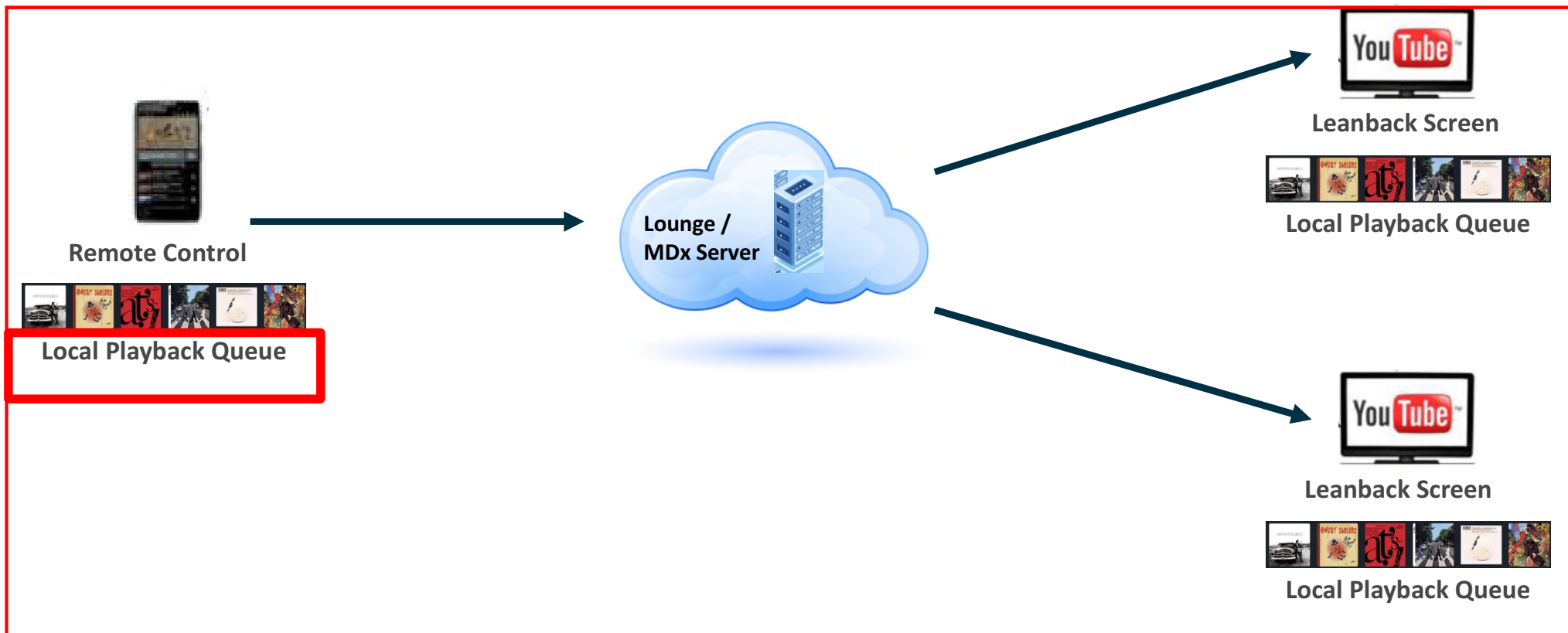# YTR DOES **NOT** DISCLOSE LIMITATION 1.4

YTR Does Not Disclose Limitation 1.4 of the '033 Patent

Limitation 1.4: operating in a first mode in which the computing device is configured for playback of a remote playback queue provided by a cloud-based computing system associated with a cloud-based media service.

___

➢ Limitation 1.4 requires the "computing device" (*e.g.*, phone) to be "configured for playback of a *remote* playback queue."

➢ Regardless of its mode of operation, the YTR System does not disclose limitation 1.4.

➢ The remote controls in the YTR System use a *local* playback queue.

➢ At the very least, there is a genuine dispute over the material facts regarding this limitation.

Rebuttal Expert Report of Douglas C. Schmidt, ¶¶171-172, 327-331

50

YTR Does Not Disclose Limitation 1.4 of the '033 Patent

➢ In non-party mode, the YTR remote control uses a local playback queue:

Remote Control

Local Playback Queue

Lounge / MDx Server

Leanback Screen

Local Playback Queue

Leanback Screen

Local Playback Queue

Demonstrative from Rebuttal Expert Report of Douglas C. Schmidt (annotated)

51

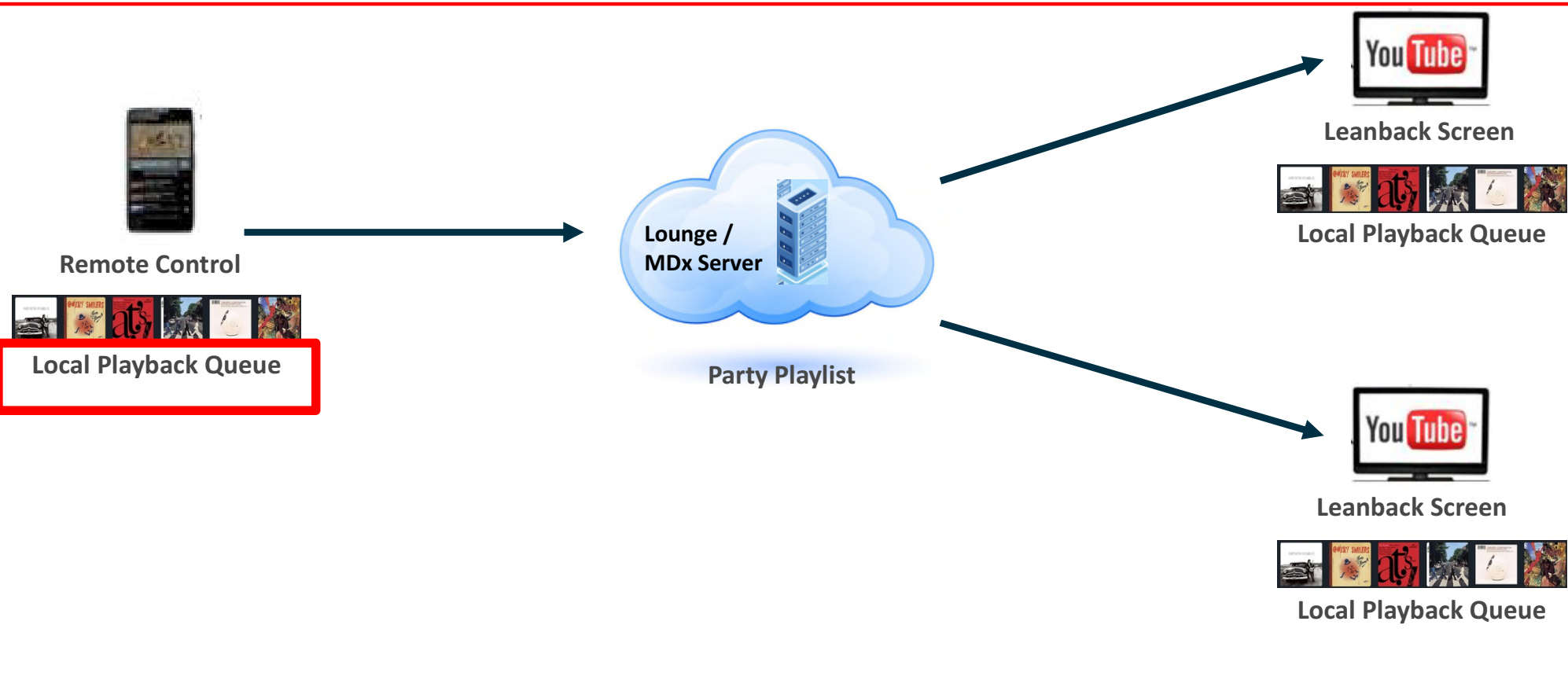# YTR Does Not Disclose Limitation 1.4 of the '033 Patent

➤ In party mode, the YTR remote control also uses a local playback queue:

Demonstrative from Rebuttal Expert Report of Douglas C. Schmidt (annotated)

52

YTR Does Not Disclose Limitation 1.4 of the '033 Patent

➢ YTR may "recommend" videos that, if desired, a user can *manually* select for playback by adding such videos (*i.e.*, pressing the plus button next to the recommended video) to the playback queue stored *locally* on the remote control:

Video #6 (GOOG-SONOSNDCA-00071318) at 1:44-1:50

YTR Does Not Disclose Limitation 1.4 of the '033 Patent

➢ In contrast, for the accused YouTube applications, the phone receives a list of "service-recommended media items" that have been selected for playback from a queue that is maintained by Google's YouTube cloud infrastructure (*i.e.*, the Watch Next queue).

➢ In other words, Google moved from (i) a system where a user manually adds desired items to a local queue, to (ii) a system in which a server delivers items stored in a cloud queue to a phone for playback.

Dkt. 509.03, ¶¶125-28, 229, 241

YTR Does Not Disclose Limitation 1.4 of the '033 Patent

➢ Google's *automatic* playback of "recommended" videos argument is new and should be stricken.

➢ Google did not raise this argument in its contentions, which creates severe prejudice for Sonos, which has never had an opportunity to take discovery about this theory.

➢ Discovery could have revealed that the cloud server providing the "service-recommended media items" in YTR did not maintain a remote playback queue, but rather, aggregated a new set of recommended videos upon each request to fill the local playback queue of the requesting remote control.

Dkt. 463.02, pp. 7-8

YTR Does Not Disclose Limitation 1.4 of the '033 Patent

➢ Google's YTR argument points to disjointed modes of operation (non-party mode's "service-recommended media items" and party mode) to construct "remote playback queues" at both the phone and the screen.

➢ There is no dispute that YTR non-party mode and party mode are separate and distinct modes of operation that cannot be run at the same time.

➢ According to Google, YTR party mode relies on a "party queue" created by a user, not a set of media items recommended by a cloud server.

Dkt. 482.12), pp. 6-7, 11-12

YTR Does Not Disclose Limitation 1.4 of the '033 Patent

➢ In short, Google has not demonstrated, with clear and convincing evidence, and without any genuine disputes over any material facts, that YTR non-party mode and YTR party mode could be used together at the same time to disclose a system that meets both limitations 1.4 and 1.7.

➢ They cannot.

IT WOULD **NOT** HAVE BEEN OBVIOUS TO ADD A DEVICE PICKER TO YTR PARTY MODE

It Would Not Have Been Obvious to Add a "Device Picker"

➢ Limitations 1.5-1.6 require "displaying a representation of one or more playback devices" and "receiving user input indicating a selection of at least one given playback device," respectively.

➢ It would not have been obvious to add such a device picker to YTR party mode.

➢ At the very least, there is a genuine dispute over the material facts regarding this limitation.

It Would Not Have Been Obvious to Add a "Device Picker"

➤ As Sonos's technical expert, Doug Schmidt, explained in his rebuttal expert report:

The purpose of party mode is to allow a host and his or her guests to playback the same media items on their Leanback Screens . . . . Thus, there would be no need to use a device-picker to select a particular Leanback Screen for playback, when the desire is to have multiple Leanback Screens for playback. Doing so would result in the host or the guests having his or her Leanback Screen being undesirably excluded from the party playback. And even if only a single Leanback Screen was being used in a party mode session with multiple remote controls, there would be no need for a device-picker in such a case. In other words, not only is a device-picker unnecessary for party mode, it would ruin it.

Rebuttal Expert Report of Douglas C. Schmidt, ¶351